**Ермаков Никита Денисович**

**ДКИП-203прог**

**Лабораторная работа 5. Одномерные массивы**

**Вариант 3**

**В одномерном массиве, состоящем из п целочисленных элементов, вычислить:**

произведение элементов массива с четными номерами;

сумму элементов массива, расположенных между первым и последним пулевыми элементами.

Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом - все отрицательные (элементы, равные нулю, считать положительными).

Код:

using System;

class Program

{

static void Main()

{

// Инициализация массива

int[] array = { 3, -1, 0, 7, -2, 5, 0, -3, 6 };

// 1. Произведение элементов массива с четными номерами

int productEvenIndex = 1;

bool hasEvenIndexElements = false;

for (int i = 0; i < array.Length; i += 2)

{

productEvenIndex \*= array[i];

hasEvenIndexElements = true;

}

if (!hasEvenIndexElements)

{

productEvenIndex = 0; // В случае отсутствия четных индексов элементов

}

Console.WriteLine($"Произведение элементов с четными индексами: {productEvenIndex}");

// 2. Сумма элементов между первым и последним нулевыми элементами

int firstZeroIndex = -1;

int lastZeroIndex = -1;

for (int i = 0; i < array.Length; i++)

{

if (array[i] == 0)

{

if (firstZeroIndex == -1)

{

firstZeroIndex = i;

}

lastZeroIndex = i;

}

}

int sumBetweenZeros = 0;

if (firstZeroIndex != -1 && lastZeroIndex != -1 && firstZeroIndex != lastZeroIndex)

{

for (int i = firstZeroIndex + 1; i < lastZeroIndex; i++)

{

sumBetweenZeros += array[i];

}

}

Console.WriteLine($"Сумма элементов между первым и последним нулевыми элементами: {sumBetweenZeros}");

// 3. Преобразование массива: положительные элементы вперед

int[] positiveElements = Array.FindAll(array, x => x >= 0);

int[] negativeElements = Array.FindAll(array, x => x < 0);

int[] transformedArray = new int[positiveElements.Length + negativeElements.Length];

Array.Copy(positiveElements, transformedArray, positiveElements.Length);

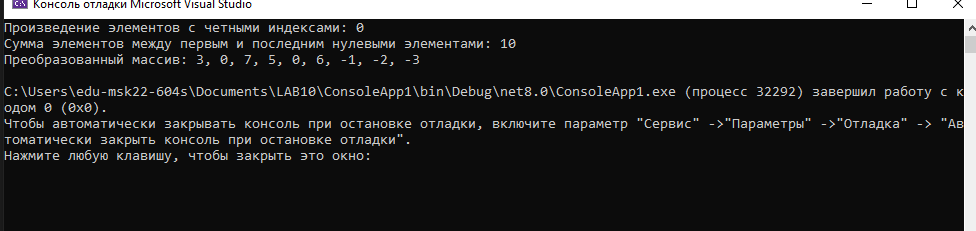
Array.Copy(negativeElements, 0, transformedArray, positiveElements.Length, negativeElements.Length);

// Вывод преобразованного массива

Console.WriteLine("Преобразованный массив: " + string.Join(", ", transformedArray));

}

}



**Лабораторная работа 6. Двумерные массивы**

**Вариант 3**

**Дана целочисленная прямоугольная матрица. Определить:**

количество столбцов, содержащих хотя бы один нулевой элемент;

помер строки, в которой находится самая длинная серия одинаковых элементов.

Код:

using System;

class Program

{

static void Main(string[] args)

{

int[,] matrix = {

{ 1, 2, 0 },

{ 4, 5, 6 },

{ 0, 8, 9 }

};

int columnCountWithZero = CountColumnsWithZero(matrix);

int longestSeriaRowLength = LongestSeriaRowLength(matrix);

Console.WriteLine($"Количество столбцов с хотя бы одним нулевым элементом: {columnCountWithZero}");

Console.WriteLine($"Длина самой длинной серии одинаковых элементов: {longestSeriaRowLength}");

}

static int CountColumnsWithZero(int[,] matrix)

{

int columnCount = matrix.GetLength(1);

bool[] hasZeroInColumn = new bool[columnCount];

for (int i = 0; i < matrix.GetLength(0); i++)

{

for (int j = 0; j < columnCount; j++)

{

if (matrix[i, j] == 0)

{

hasZeroInColumn[j] = true;

}

}

}

int count = 0;

foreach (bool hasZero in hasZeroInColumn)

{

if (hasZero) count++;

}

return count;

}

static int LongestSeriaRowLength(int[,] matrix)

{

int longestLength = 0;

for (int i = 0; i < matrix.GetLength(0); i++)

{

int currentLength = 1;

for (int j = 1; j < matrix.GetLength(1); j++)

{

if (matrix[i, j] == matrix[i, j - 1])

{

currentLength++;

}

else

{

longestLength = Math.Max(longestLength, currentLength);

currentLength = 1;

}

}

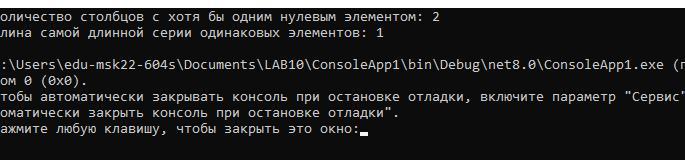
longestLength = Math.Max(longestLength, currentLength);

}

return longestLength;

}

}

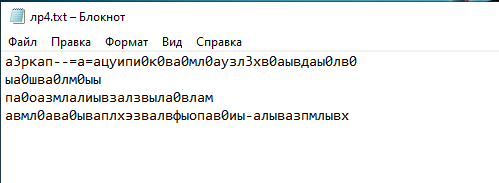


**Лабораторная работа 7. Строки**

**Вариант 2**

Написать программу, которая считывает текст из файла и выводит на экран только предложения, содержащие введенное с клавиатуры слово.

Файл со словом:



Код:

using System;

using System.IO;

using System.Linq;

class Program

{

static void Main(string[] args)

{

// Запрос имени файла

Console.Write("Введите путь к файлу: ");

string filePath = Console.ReadLine();

// Проверка существования файла

if (!File.Exists(filePath))

{

Console.WriteLine("Файл не найден. Пожалуйста, проверьте путь к файлу.");

return;

}

// Чтение содержимого файла

string content = File.ReadAllText(filePath);

// Запрос слова для поиска

Console.Write("Введите слово для поиска: ");

string searchTerm = Console.ReadLine();

// Разделение текста на предложения

var sentences = content.Split(new[] { '.', '!', '?' }, StringSplitOptions.RemoveEmptyEntries)

.Select(sentence => sentence.Trim());

// Вывод предложений, содержащих искомое слово

Console.WriteLine("Предложения, содержащие слово \"{0}\":", searchTerm);

foreach (var sentence in sentences)

{

if (sentence.IndexOf(searchTerm, StringComparison.OrdinalIgnoreCase) >= 0)

{

Console.WriteLine(sentence);

}

}

}

}

